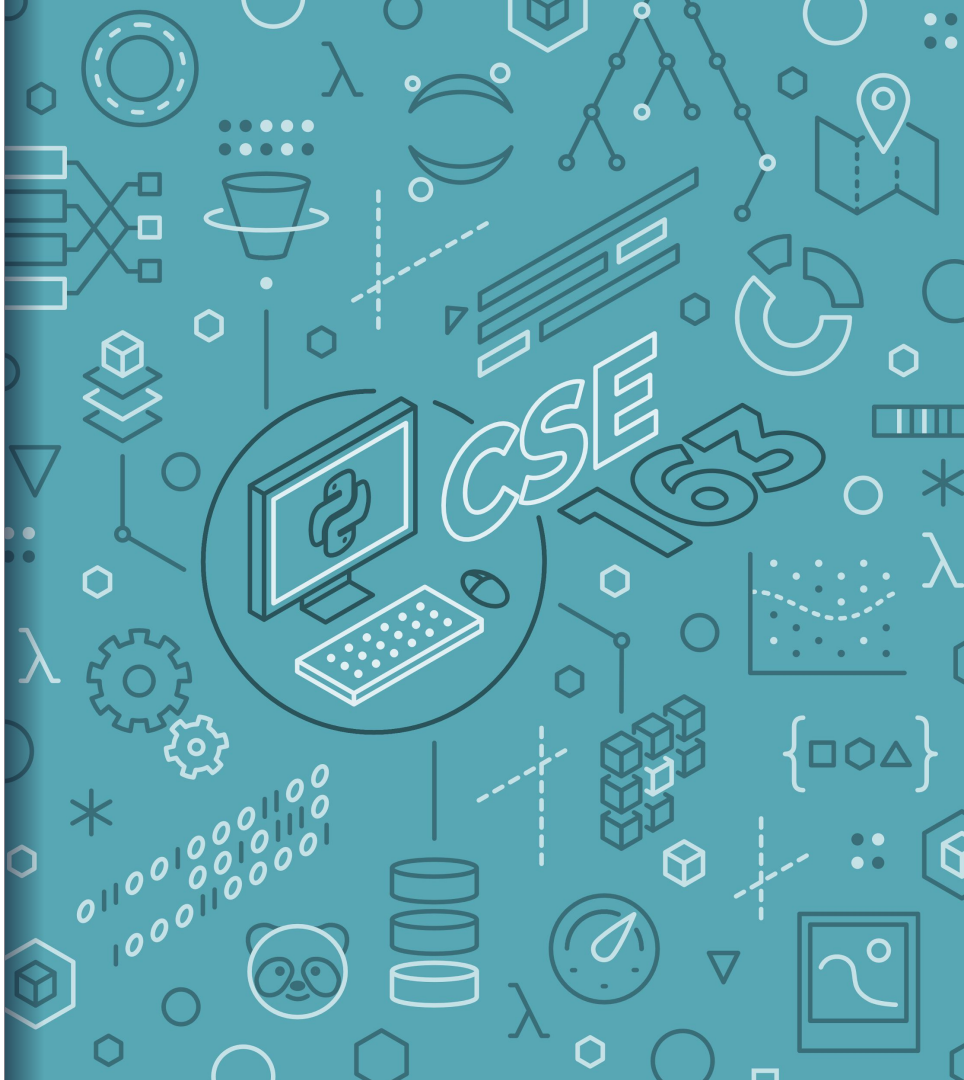


CSE 163

More Pandas

Hunter Schafer

DataFrame

- One of the basic data types from pandas is a DataFrame
  - It's essentially a table with column and rows!

Columns

Index (row)

	id	year	month	day	latitude	longitude	name	magnitude
0	nc72666881	2016	7	27	37.672333	-121.619000	California	1.43
1	us20006i0y	2016	7	27	21.514600	94.572100	Burma	4.90
2	nc72666891	2016	7	27	37.576500	-118.859167	California	0.06

Location

- So far, we have shown you how to access columns and filter

Series

```
series[<indexer>]
```

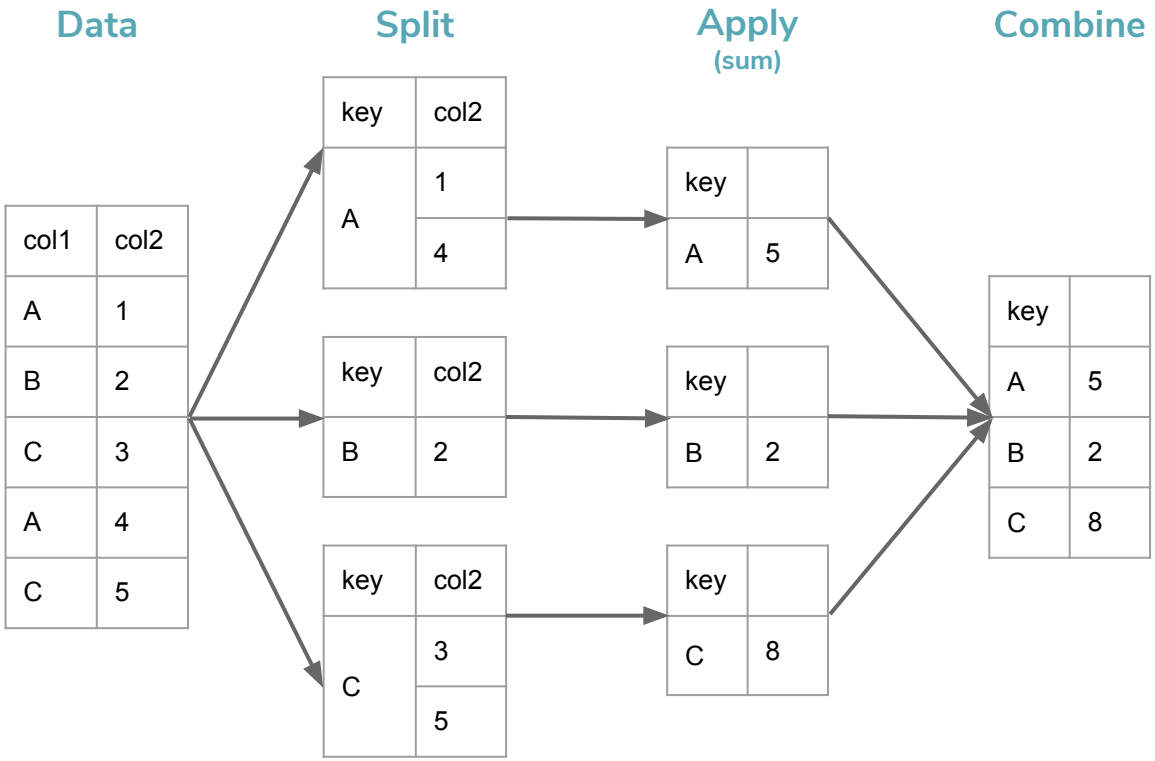
DataFrame

```
data.loc[<row indexer>, <column indexer>]
```

- Location Demo

Group By

```
data.groupby('col1')['col2'].sum()
```



Poll Everywhere

Think

1.5 minutes

pollev.com/cse163

What pandas call would we write to solve the following problem?

Compute the largest magnitude earthquakes that happened in 2016 at each named place above the equator in a DataFrame called d.

	id	year	month	day	latitude	longitude	name	magnitude
0	nc72666881	2016	7	27	37.672333	-121.619000	California	1.43

A)
```
d[d['lat'] >= 0 and d['year'] == 2016].groupby('name')['mag'].max()
```

B)
```
d[d['lat'] >= 0 & d['year'] == 2016].groupby('name')['mag'].max()
```

C)
```
d[(d['lat'] >= 0) & (d['year'] == 2016)].groupby('name')['mag'].max()
```

D)
```
d.groupby('name')['mag'].max()[d['lat'] >= 0 & d['year'] == 2016]
```

E)
```
d.groupby('name')['mag'].max()[(d['lat'] >= 0) & (d['year'] == 2016)]
```

Poll Everywhere

Pair

3 minutes!

pollev.com/cse163

What pandas call would we write to solve the following problem?

Compute the largest magnitude earthquakes that happened in 2016 at each named place above the equator in a DataFrame called d.

	id	year	month	day	latitude	longitude	name	magnitude
0	nc72666881	2016	7	27	37.672333	-121.619000	California	1.43

A)
```
d[d['lat'] >= 0 and d['year'] == 2016].groupby('name')['mag'].max()
```

B)
```
d[d['lat'] >= 0 & d['year'] == 2016].groupby('name')['mag'].max()
```

C)
```
d[(d['lat'] >= 0) & (d['year'] == 2016)].groupby('name')['mag'].max()
```

D)
```
d.groupby('name')['mag'].max()[d['lat'] >= 0 & d['year'] == 2016]
```

E)
```
d.groupby('name')['mag'].max()[(d['lat'] >= 0) & (d['year'] == 2016)]
```

Looping

- Rarely, you will need to loop over your DataFrame
- This is not advised since it is slow to write a loop in Python to operate on your DataFrame
  - When we are running Python, it is being **interpreted**
- Generally want the DataFrame to do all the work for us
  - DataFrames are fast because they are written in C

```
# To get the values
for val in series:
    print(val)

# To get the indices
for i in series.index:
    print(i, series[i])
```

Apply

- We have shown how to filter and group your data, but sometimes you want to transform your data
- Pretty easy to change numerical data using the operators we learned last time (+, -, /, *, abs(), min(), max(), etc.)
- With Strings, it's not so easy

```
data['name'].str.len()
data['name'].str.upper()
data['name'].apply(len)
data['name'].apply(my_function)
```

- The last two pass a function as a parameter!

Apply Demo

Advice from HW1

- Start early
- Write tests as you go

Next Week

- All about data science
  - Time series data
  - Data visualization
  - Machine learning

Before Next Time

- Keep up with practice!
- Start HW2 early!
- Will post Project deadlines and finalized exam dates this weekend!